What is claimed is:

1. A system, comprising:

a memory space;

a task;

a first task control block associated with the task and located in a first area of the memory space, the first task control block including a number of first task information data structures that contain first task information;

a second task control block associated with the task and located in a second area of the memory space, the second task control block including a number of second task information data structures that contain second task information;

wherein the first area of the memory space is not directly accessible by the task, and the second area of the memory space is directly accessible by the task.

2. The system of claim 1, further comprising a current task data structure, and wherein the current task data structure is loaded with information from at least one of the first task information data structures during a context switch to execute the task.

3. The system of claim 2, further comprising a context switch routine, the context switch routine performing the context switch to execute the task.

4. The system of claim 1, wherein the first task control block includes a pointer data structure that contains a pointer to a location of the second task control block.

5. The system of claim 1, wherein the first area of the memory space is a system space and the second area of the memory space is a user space.

6. The system of claim 1, wherein the number of second task information data structures includes error status information for the task.

7. The system of claim 1, wherein the number of second task information data structures includes a set of pointers to a set of standard modules for the task.

8. The system of claim 1, wherein the number of second task information data structures includes a pointer to environment variables for the task.

9. The system of claim 1, wherein the number of second task information data structures includes a pointer to context information for remote procedure calls made by the task.

10. The system of claim 1, wherein the number of second task information data structures includes a pointer to context information for remote procedure calls made by the task.

11. The system of claim 1, wherein the number of second task information data structures includes a pointer to exception information.

12. The system of claim 1, wherein the number of second task information data structures includes a user-definable spare field.

13. The system of claim 1, further comprising a real-time operating system.

14. A method, comprising:

receiving a request to create a task;

assigning task information for the task;

creating a first task control block for the task;

loading the task information for the task into the first task control block;

creating a second task control block for the task, the second task control block having a location in a memory space;

loading an address for the location of the second task control block into the first task control block.

15. The method of claim 14, wherein the first task control block is located in a system space and the second task control block is located in a user space, the system space and user space both within the memory space.

16. The method of claim 14, wherein the first task control block and the second task control block are located in a system space, the system space within the memory space.

17. A method, comprising:

receiving a context switching event;

saving task information for a first task in a system task control block associated with the first task, the task information for the first task including a pointer to a user task control block associated with the first task;

loading task information for a second task from a system task control block associated with the second task, the task information for the second task including a pointer to a user task control block associated with the second task.

18. A method, comprising:

receiving an interrupt request;

saving a first number of state information values from a current task data structure for a currently executing task, the current task data structure including a pointer data structure holding a pointer to a second number of state information values;

loading a pointer to a task control block associated with an interrupt service routine into the pointer data structure;

executing the interrupt service routine.

19. The method of claim 18, further comprising:

creating the task control block associated with the interrupt service routine.

20. The method of claim 18, wherein the first number of state information values are saved on an interrupt stack.

21. The method of claim 20, wherein the second number of state information values are not saved on the interrupt stack.